CLAIMS

We claim:

1. A data transformation system comprising:

a data interface configured to receive data to be transformed or to send transformed data;

memory configured to store one or more transform process definitions having at least one simple transform definition and at least one compound transform definition;

an application including computer instructions; and

a data interpreter configured to exchange data with the data interface and the application, the data interpreter including a transform engine configured to

select a transform process definition from the one or more transform process definitions, the selected transform process definition including a hierarchical data structure,

concurrently navigate the selected transform process definition and the data to be transformed, navigation within the data to be transformed being response to transform definitions within the selected transform process definition, and

generate output data having a data structure responsive to a data structure of the selected transform process definition.

2. The data transformation system of claim 1, wherein the transformation engine is further configured to process the at least one compound transform definition using recursion.

3. The data transformation system of claim 1, wherein the data interpreter is further configured to support a plurality of applications.

4. The data transformation system of claim 1, wherein the application is a database application, accounting application, human resources application, customer management application, inventory application, or an internet application.

5. The data transformation system of claim 1, wherein the application and the data interpreter are integrated.

6. The data transformation system of claim 1, wherein the data interpreter further includes a computing device configured to support the transform engine.

7. A data interpreter configured to transform data to be transformed, the data interpreter comprising:

at least one computing device; and

a transform engine supported by the computing device, the transform engine being configured to

access a transform process definition including a hierarchical data structure of transform definitions, the data structure including a simple transform definition and a compound transform definition, concurrently navigate the transform process definition and the data to be transformed, navigation within the data to be transformed being response to the transform definition within the transform process definition, and

generate output data having a data structure responsive to the transform process definition.

8. The data interpreter of claim 7, wherein the data structure of the output data is responsive to a structure of transform process definition.

9. The data interpreter of claim 7, wherein the transform engine is configured to process the compound transform definition using recursion.

10. The data interpreter of claim 7, wherein the transform engine is configured to generate output data including data elements characterized by the transform process definition and having no contribution from the data to be transformed.

11. The data interpreter of claim 7, wherein the transform process definition is configured such that some data elements in the data to be transformed do not make a contribution to the output data.

12. The data interpreter of claim 7, wherein the transform engine is further configured to navigate the data to be transformed responsive to the date structure of the transform definitions within the transform process definition.

13. The data interpreter of claim 7, wherein the transform engine is further configured to navigate the data to be transformed responsive to content of the transform definitions.

14. A method of transforming data using an application programming interface, the method comprising:

receiving data to be transformed at the application programming interface;

parsing identification data within the data to be transformed, the identification data characterizing the data to be transformed;

using the identification data to select a transform process definition from a set of one or more transform process definitions, the selected transform process definition defining a process of translating data elements within the data to be transformed to output data elements; and

transforming the data to be transformed to output data, using a transformation engine and the selected transform process definition, a data structure of the output data being responsive to a data structure of the transform process definition.

15. The method of claim 14, wherein transforming the data to be transformed includes nesting of data records.

16. The method of claim 14, wherein transforming the data to be transformed includes calling a transformation process recursively responsive to a compound transform definition in the selected transform process definition.

17. The method of claim 14, wherein the selected transform process definition is selected based on information within the identification data that identifies a destination of the data to be transformed.

18. The method of claim 14, wherein the transform process definition is selected based on information within the identification data that identifies a format of the output data.

19. The method of claim 14, wherein the transform process definition includes a extensible markup language (XML).

20. A method of transforming data using an application programming interface, the method comprising:

receiving data to be transformed at the application programming interface, the data to be transformed including identification data;

using the identification data to select a transform process definition from a set of transform process definitions, the selected transform process definition defining a process of translating data elements within data to be transformed to output data elements; and

transforming the data to be transformed to output data by concurrently navigating the data to be transformed and the selected transform process definition, navigation in the data to be transformed being responsive to the transform process definition.

21. The method of claim 20, wherein the selected transform process definition is selected based on information within the identification data that identifies a format of the data to be transformed.

22. The method of claim 20, wherein the selected transform process definition is selected based on information within the identification data that identifies a source of the data to be transformed.

23. The method of claim 20, wherein the application programming interface is shared by several applications.

24. The method of claim 20, wherein the application programming interface is shared by several applications and the selected transform process definition is selected based on an identity of one of the several applications.

25. The method of claim 20, wherein a structure of the output data is responsive to a structure of a transform definition included in the selected transform process definition.

26. The method of claim 20, wherein transforming the data to be transformed includes nesting of data records.

27. The method of claim 20, wherein transforming the data to be transformed includes filtering of data records.

28. The method of claim 20, wherein transforming the data to be transformed includes calling a transformation process recursively responsive to a data structure of a transform definition included in the selected transform process definition.

29. The method of claim 20, wherein transforming the data to be transformed includes searching the data to be transformed for a data field specified in a transform definition included in the transform process definition.

30. The method of claim 20, wherein the transform definition includes a translation codeset parameter configured to invoke an external reference.

31. The method of claim 20, wherein the transform definition includes a translation codeset parameter configured to invoke an external reference, the external reference being configured to perform logic operations using the data to be transformed.

32. A method of transforming data, the method comprising:

positioning a definition pointer to point at one of a plurality of transform definitions within a transform process definition;

reading the pointed at transform definition;

searching_data to be transformed for a data element to be transformed, the search being responsive to the pointed at transform definition; and transforming any found data element into output data, responsive to the pointed at transform definition, a data structure of the output data being responsive to a data structure of the transform process definition.

33. The method of claim 32, further including determining a type of the read transform definition and, if the transform definition is not a simple transform definition type, recursively calling the method of claim 32.

34. The method of claim 32, further including determining if all sub-definitions of a compound transform definition have been processed.

35. The method of claim 32, wherein the method of transforming data includes nesting of a data element.

36. The method of claim 32, further including, if no data element is found in the step of searching data to be transformed, adding an output data element to the output data responsive to the read transform definition, the data to be transformed having no contribution to the output data element.

37. The method of claim 32, wherein the read transform definition includes a value parameter configured to specify a value for inclusion in the output data.

38. The method of claim 32, wherein the data element is a compound data element and the read transform definition includes a source record parameter configured to specify the compound data element.

39. The method of claim 32, wherein the read transform definition is in a meta-language format.

40. The method of claim 32, wherein the data to be transformed data is in a meta-language data format.

41. The method of claim 32, wherein the read transform definition includes a transform element having an output field name and a source field parameter.

42. The method of claim 32, wherein the read transform definition includes a value parameter configured to populate a field in the output data.

43. The method of claim 32, wherein the read transform definition includes a translation codeset parameter.

44. A method of transforming data, the method comprising:

positioning a definition pointer to point at a transform definition, the transform definition being one of a plurality of transform definitions within a transform process definition;

reading the pointed at transform definition;

positioning a payload pointer to point at a data element to be transformed, the positioning being responsive to a data structure of the transform process definition; and

transforming the data element into output data, responsive to the read transform definition.

45. The method of claim 44, further including determining a type of the read transform definition and, if the read transform definition is not a simple transform definition type, recursively calling the method of claim 44.

46. The method of claim 44, further including determining a type of the read transform definition and, if the read transform definition is not a simple transform definition type, recursively calling the method of claim 44, wherein the recursive call is responsive to the data structure of the transform process definition.

47. The method of claim 44, further including determining a type of the read transform definition, if the read transform definition is not a simple transform definition type recursively calling the method of claim 44, and determining if all sub-elements of a compound element have been transformed.

48. The method of claim 44, further including determining if all sub-elements of a compound element have been transformed and, if the determination returns a value of YES, returning to a calling process.

49. The method of claim 44, further invoking a parallel process to process sub-definitions of the read transform definition.

50. The method of claim 44, wherein the method of transforming data includes un-nesting of the data element to be transformed.

51. The method of claim 44, wherein the read transform definition includes a source field parameter configured to specify the data element.

52. The method of claim 44, wherein the read transform definition includes a source record parameter configured to specify the compound data element.

53. The method of claim 44, wherein the read transform definition includes a translation codeset configured for calling computer instructions including logic operations.

54. The method of claim 53, wherein the computer instructions are configured to call an external process.

55. The method of claim 44, further including a step of combining the data element with the transform process definition prior to transforming the data element to output data.

56. The method of claim 44, wherein the transform process definition includes a tree data structure.

57. A method of preparing data for transformation, the method comprising:

receiving data to be transformed;

parsing the received data to determine identification information;

using the identification information to extract a transform process definition from a plurality of transform process definitions, the extracted transform process definition including a transform definition configured to transform the data to be transformed, to direct navigation within the data to be transformed during transformation, and to determine a data structure of output data resulting from transformation of the data to be transformed, the transform definition including a hierarchical data structure having at least one simple transform definition and at least one compound transform definition, the compound transform definition being configured to generate a compound data element in the output data; and

adding the extracted transform process definition to meta-language transform input data including the data to be transformed.

58. The method of claim 57, wherein the extracted transform process definition is in a meta-language format.

59. A computer readable media having embodied thereon data, the data comprising:

computer instructions configured to position a definition pointer to point at a transform definition, the transform definition being one of a plurality of transform definitions within a transform process definition;

computer instructions configured to read the pointed at transform definition;

computer instructions configured to increment a payload pointer, within the data to be transformed, to a data element to be transformed, the incrementation being responsive to the pointed at transform definition; and

computer instructions configured to transform any found data element into output data, responsive to the pointed at transform definition, a data structure of the output data being responsive to a data structure of the transform process definition.

60. The computer readable media of claim 59, wherein the data further comprises computer instructions configured to employ recursion to transform a compound data element within the data to be transformed.

61. The computer readable media of claim 59, wherein the data further comprises computer instructions configured to transform the data to be transformed using parallel processes.

62. A computer readable media having embodied thereon data, the data comprising:

payload data including data to be transformed, the data to be transformed including metadata characterizing simple data elements and compound data elements; and

a transform process definition including a transform definition configured to transform the data to be transformed, to direct navigation within the data to be transformed during transformation, and to determine a data structure of output data resulting from the transformation, the transform definition including a hierarchical data structure having at least one simple transform definition and at least one compound transform definition, the compound transform definition being configured to generate a compound data element in the output data.

63. The computer readable media of claim 62, wherein the computer readable media includes memory included in a data interface.

64. The computer readable media of claim 62, wherein the computer readable media includes a hard drive.

65. An application system comprising:

means for positioning a definition pointer to point at a transform definition within a transform process definition;

means for reading the transform definition;

means for positioning a payload pointer to point to a first data element, the first data element being a member of a plurality of data elements within data to be transformed; and

means for generating output data using the first data element and the transform definition.

66. The application system of claim 65, further including means for selecting the transform process definition from a set of transform process definitions, responsive to data associated with the data to be transformed.

67. The application system of claim 65, wherein a second data element has no contribution to output data generated using the transform process definition, the second data element being a member of the plurality of data elements.

68. The application system of claim 65, further including means for adding data to the output data, the added data being configured responsive to the transform process definition and having no contribution from the data to be transformed.